# A new iterative heuristic to solve the Joint Replenishment Problem using a spreadsheet technique

† Andreas Nilsson*
† Anders Segerstedt
‡ Erik van der Sluis

† Luleå *University of Technology, Division of Industrial Logistics, S-97187 Luleå, Sweden*
‡ *University of Amsterdam, Faculty of Economics and Econometrics, Roetersstraat 11, 1018 WB Amsterdam, The Netherlands*

* *Corresponding Author: Andreas.n.nilsson@ltu.se*

**Abstract**

In this paper, a heuristic method is presented which gives a novel approach to solve joint replenishment problems (JRP) with strict cycle policies. The heuristic solves the JRP in an iterative procedure and is based on a spreadsheet technique. The principle of the recursion procedure is to find a balance between the replenishment and inventory holding costs for the different items by adjusting the replenishment frequencies. The heuristic is also tested according to an extensive test template and shows pleasing results. It also performs well in comparison with many other heuristics.

*Keywords:* Inventory; Joint replenishment; Heuristics

## 1. Introduction

The joint replenishment problem (JRP) is a famous real-world problem that occurs in different situations. For instance when; 1) Several items are ordered from a single supplier; 2) Several products share the same means of transportation; 3) One product is packaged after being manufactured in different quantities.

The characteristic of the JRP is two types of replenishment costs. One major replenishment cost when any of the items are ordered and a minor, individual replenishment cost when a particular item is ordered. In addition, there is also an individual inventory holding cost involved as well. The structure of an optimal strategy can be very complex, even for systems with only a few products. For all items, stock-outs are not allowed and demand is known and constant.

Different strategies are proposed for the JRP according to direct grouping strategies (DGS) and indirect grouping strategies (IGS). In the first strategy (DGS), items are divided into different groups where all items in each group are jointly replenished. Each group has its own cycle time. With the latter strategy (IGS), joint replenishment opportunities are scheduled at constant intervals of time and the quantity ordered of each item is sufficient to last for an integer multiple of the *basic interval*. A replenishment opportunity does not have to be used. We will in this paper focus on strict cycle policies, which means that we assume that at least one item is ordered at every replenishment opportunity. This makes the problem less abstract for the user of such an algorithm and in most of the cases, the optimal solutions are the same using either cycle policies or strict cycle policies.

The work on methods to solve the joint replenishment problem throughout the years has been extensive. On optimal deterministic models, readers are referred to the publications of Fung and Ma (2001), Goyal (1974, 1988a), Van Eijs (1993), Viswanathan (1996, 2002). Wildeman et al. (1997). There exists also several heuristic methods of interest, some complemented with technical notes. See among other publications: Brown (1967), Goyal (1973, 1985, 1988b), Goyal and Belton (1979), Goyal and Deshmukh (1993), Hariga (1994a,b), Kaspi and Rosenblatt (1983, 1985, 1991), Nocturne (1973), Olsen (2005) and Silver (1976).

The common feature of most heuristics presented in literature is the iterative way for obtaining a solution. Given a time interval for the joint replenishments, optimal order frequencies are determined. Given these order frequencies a new time interval is determined. These steps are repeated until the solution converges. In this paper we will propose a new method where we balance the replenishment- and inventory holding costs for the different items in an iterative procedure.

The paper has the following outline. Firstly general notation is described and two necessary formulas for the heuristics are derived. Then, the principle of the model and the recursion procedure are described. We start by presenting a schematic recursion procedure simply described in words, followed by a detailed recursion procedure and ending by giving an illustrated numerical example. The heuristic is then tested according to an extensive test template and the results are compared with earlier heuristics in terms of closeness to optimality, average and maximum error. Finally, we present some conclusions.

## 2. Notation and important equations for our heuristic method

| | |
|---|---|
| $d_i$ | Demand rate in units per unit time for item *i* |
| $A$ | Major replenishment cost for the family of items |
| $a_i$ | Minor replenishment cost for item *i* |
| $h_i$ | Inventory holding cost per unit and time unit for item *i* |
| $N$ | Number of items in the family |
| $T$ | Time interval in units within which at least one item has to be replenished |
| $m_i$ | The integer number of *T* intervals that the replenishment quantity of item *i* will last. |
| $\mathbf{m} = (m_1,.,m_N)$ | Vector containing **m**-values for all items |
| $C(\mathbf{m})$ | Minimal total cost function given **m** |
| $Q_i(\mathbf{m})$ | Quotient between replenishment cost and holding cost for item *i* given **m** |
| $\mathbf{\Theta}_K$ | Subset of {1, 2, …, *N*} containing *K* items where $K \in \{1, 2,..., N\}$ |

The total cost function expressed per time unit is the following:

$$C(T;\mathbf{m}) = \frac{A}{T} + \sum_{i}^{N}\left(\frac{a_i}{Tm_i} + \frac{Tm_i d_i h_i}{2}\right) \quad (1)$$

The optimal time interval, $T^*$, that minimizes Eq. (1) with a given set of $m_1, m_2, ..., m_N$ is easily found by derivation.

$$T^*(\mathbf{m}) = \sqrt{2(A + \sum_{i}^{N}\frac{a_i}{m_i}) \Big/ \sum_{i}^{N} m_i d_i h_i} \quad (2)$$

Substituting back optimal $T^*$ into Eq. (1) gives the total cost function, which depends only on the set of **m**-values. This is the first of only two formulas needed for the heuristics.

$$C(\mathbf{m}) = \sqrt{2\left(A + \sum_{i}^{N}\frac{a_i}{m_i}\right)\left(\sum_{i}^{N} m_i d_i h_i\right)} \quad (3)$$

The replenishment cost and inventory holding cost for item *i*.

$$C_i^a = \frac{a_i}{Tm_i} \quad (4)$$

$$C_i^h = \frac{Tm_i d_i h_i}{2} \quad (5)$$

Dividing Eq. (4) with Eq. (5) gives the quotient or the ratio between the two costs for item *i*.

$$Q_i = \frac{2a_i}{T^2 m_i^2 d_i h_i} \quad (6)$$

Substituting back of the optimal $T^*$(Eq. 2.) into Eq. 6 gives the quotient formula for item *i* depending only on the set of **m**-values. This is the second formula needed for the heuristic.

$$Q_i(\mathbf{m}) = \left(\sum_{i}^{N} m_i d_i h_i \Big/ (A + \sum_{i}^{N}\frac{a_i}{m_i})\right)\frac{a_i}{m_i^2 d_i h_i} \quad (7)$$

### 3. Principle of the model and recursion procedure

Segerstedt (1999) presented a heuristic method to solve an Economic Lot Scheduling Problem (ELSP). The ELSP problem occurs when several products are processed in a capacity constrained facility, where only one product can be processed at the same time. This heuristic has been modified and reworked to better suit the JRP-problem, but still has the same basic principle to balance the replenishment cost and the holding cost for each individual item. In the original heuristic the frequencies are changed individually and local optimal solutions are found by recalculating the cycle-time.

When using the traditional Economic Order Quantity (EOQ) for an individual item, the quotient or the ratio between the replenishment cost and the inventory holding cost without any safety stock is equal to one. The further away the quotient is from one, the higher the cost. Also, for any value of the quotient, the inverted value (1 / quotient) presents the same total cost. It is simply a matter of keeping a balance between the replenishment and inventory holding cost.

Returning to the JRP-problem, the same applies; the closer the individual quotients are to one, the better the solution. It is possible to solve joint replenishment problems by adjusting the quotients to come closer to one. This will be done in a two-step heuristic, where the starting solution is when all items are replenished every time interval (all *m*-values are set to one). During both steps we simply look at the quotients and track how the total cost changes as the replenishment frequencies (*m*-values) are updated.

**Schematic recursion procedure**

*Step 0*. Set all *m*-values to 1 and compute the total cost (Eq. 3) for the initial solution.

*Step 1*. Compute quotients (Eq. 7) and increase the *m*-value(s) by one for all items with quotients higher than *1.4*. Calculate the total cost. Repeat that until all quotients are below *1.4* or the total cost starts to increase. If all quotients are below *1.4* go to step 2 or if the total cost increases, step back one step to the best solution and then go to step 2.

*Step 2*. Calculate quotients and rank them how far away they are from one. Then individually increase/decrease the *m*-value of the item with the highest ranking (furthest away from one). Calculate the total cost. Repeat all that until the total cost starts to increase, then step back one step to the best solution and try to adjust the *m*-value of the item having the second highest ranking. Repeat this step until no more items exist to examine, then go to step 3. Note, that all m-values most be $\geq 1$. If an item has the highest ranking, a quotient below one and $m = 1$ it must be skipped.

Step 3. Final solution.

Since there are several local optimal solutions from the start solution all the way to the final solution it is necessary in this heuristic to both change the frequencies together and individually. That is why the heuristic is divided into two steps.

Our objective with the research has been to come up with a simple method to solve joint replenishment problems. To make the method as simple as possible we intended to find a general value, a well-functioning compromise for step 1 that will work well in any situation. We have tested different values and found the value *1.4* to be the best compromise, see section 4.

**Detailed recursion procedure**

**Step 0**

Set $m_i = 1 \forall i$

$K = N$

Compute $C_0 = C(\mathbf{m})$

Go to Step 1

**Step 1**

Compute $Q_i(\mathbf{m}) \forall i$

$\forall i$ with $Q_i(\mathbf{m}) > 1.4$ then $m_i' \leftarrow m_i + 1$

If $C(\mathbf{m}') < C_0$ then $C_0 \leftarrow C(\mathbf{m}')$, $\mathbf{m} \leftarrow \mathbf{m}'$ and repeat step 1

Else go to step 2

**Step 2**

$k \leftarrow \arg\max_{i \in \mathbf{\Theta}_K} (Q_i, 1/Q_i)$

If $Q_k < 1$ and $m_k = 1$ then $\mathbf{\Theta}_K \leftarrow \mathbf{\Theta}_K / \{k\}, K \leftarrow K - 1$ and $k \leftarrow \arg\max_{i \in \mathbf{\Theta}_K} (Q_i, 1/Q_i)$

$\mathbf{m} \leftarrow \mathbf{m}'$; If $Q_k > 1$ then $m_k' \leftarrow m_k + 1$ else $m_k' \leftarrow m_k - 1$

If $C(\mathbf{m}') < C_0$ then $C_0 \leftarrow C(\mathbf{m}')$, $\mathbf{m} \leftarrow \mathbf{m}'$, $K \leftarrow N$ and Compute $Q_i(\mathbf{m}) \forall i$

Else $\mathbf{\Theta}_K \leftarrow \mathbf{\Theta}_K / \{k\}$ and $K \leftarrow K - 1$

If $K > 0$ then repeat step 2

Else go to Step 3

**Step 3**

$C_0$ is the best solution.

**Numerical illustration**

Consider the following example with seven items, summarized in Table 1.

Table 1
Initial data

| Items | | *A* | *B* | *C* | *D* | *E* | *F* | *G* |
|---|---|---|---|---|---|---|---|---|
| Demand rate | $d_i$ | 2500 | 300 | 700 | 225 | 650 | 150 | 100 |
| Minor replenishment cost | $a_i$ | 20,00 | 4,50 | 15,00 | 22,00 | 15,00 | 14,00 | 9,50 |
| Inventory holding cost | $h_i$ | 3,00 | 2,50 | 2,00 | 5,00 | 1,00 | 2,00 | 1,50 |
| Major replenishment cost | 30 | | | | | | | |

In table 2 a printout from a simple spreadsheet model is given. The starting solution (Step 0) when all items are replenished every replenishment cycle. By checking the quotients we see that items *D*, *E*, *F* and *G* all have a quotient higher than *1.4*. We therefore increase their *m*-values by one.

Table 2
Start solution

| | | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|
| C | Run-out time in T | $m_i$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1757.13 | Quotient (ratio) | $Q_i$ | 0.24 | 0.55 | 0.98 | **1.79** | **2.11** | **4.26** | **5.79** |
| | Ranking of quotients | | 3 | 5 | 7 | 6 | 4 | 2 | 1 |

After increasing the *m*-values for item *D*, *E*, *F* and *G* the total cost has dropped. Now, items *C*, *F* and *G* have quotients above 1.4. We continue to increase their *m*-values.

Table 3
Second iteration

| | | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|
| C | Run-out time in T | $m_i$ | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| 1677.19 | Quotient (ratio) | $Q_i$ | 0.38 | 0.85 | **1.51** | 0.69 | 0.82 | **1.65** | **2.24** |
| | Ranking of quotients | | 1 | 7 | 4 | 5 | 6 | 3 | 2 |

The increased *m*-value of items *C*, *F* and *G* did not result in a better solution and we need to step back to the former solution, the best solution so far.

Table 4
Third iteration

| | | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|
| C | Run-out time in T | $m_i$ | 1 | 1 | 2 | 2 | 2 | 3 | 3 |
| 1678.64 | Quotient (ratio) | $Q_i$ | 0.48 | 1.08 | 0.48 | 0.88 | 1.04 | 0.94 | 1.27 |
| | Ranking of quotients | | 1 | 5 | 2 | 4 | 7 | 6 | 3 |

Once again having this solution, we go to step 2 of the recursion procedure and look how far away the quotients are from one. In this case item *A* has the quotient that is furthest away from one. However, since the *m*-value of item *A* cannot be decreased ($m_i \neq 0$), we look at the second highest quotient, which belongs to item *G*. We increase the *m*-value for *G*.

Table 5
Back to best solution so far

| | | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|
| C | Run-out time in T | $m_i$ | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| 1677.19 | Quotient (ratio) | $Q_i$ | 0.38 | 0.85 | 1.51 | 0.69 | 0.82 | 1.65 | **2.24** |
| | Ranking of quotients | | 1 | 7 | 4 | 5 | 6 | 3 | 2 |

Since the total cost continued to decrease we continue the procedure of step 2. Again, item *A* is not valid, but item *F* is.

Table 6.
Fourth iteration.

| | | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|
| *C* | Run-out time in T | $m_i$ | 1 | 1 | 1 | 2 | 2 | 2 | 3 |
| 1672.65 | Quotient (ratio) | $Q_i$ | 0.39 | 0.87 | 1.56 | 0.71 | 0.84 | **1.69** | 1.02 |
| | Ranking of quotients | | 1 | 6 | 3 | 4 | 5 | 2 | 7 |

After increasing the *m*-value of item *F* the total cost drops. If we continue to change the m-value of item C the total cost will start to increase. To guarantee the best solution of the heuristic we need to try to increase the *m*-value for the item with the second, third, forth etc. highest quotient. Finally we stop when we have tried all items. In this case that will not give us any better solution. Hence, the solution in table 7 is the best solution our procedure can find, and also in this case the optimal solution of the problem (Can be verified using for instance Goyal (1988a).

Table 7.
Final solution.

| | | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|
| *C* | Run-out time in T | $m_i$ | 1 | 1 | 1 | 2 | 2 | 3 | 3 |
| 1669.96 | Quotient (ratio) | $Q_i$ | 0.40 | 0.91 | 1.63 | 0.74 | 0.88 | 0.79 | 1.07 |
| | Ranking of quotients | | 1 | 6 | 2 | 3 | 5 | 4 | 7 |

## 4. Test of the model

The testing of the new heuristic has two purposes. 1) To evaluate the performance of the heuristic 2) To find the general value for step 1.

Kaspi and Rosenblatt (1991), Goyal and Deshmukh (1993) and also partly Hariga (1994b) have all used an extensive test template to test their heuristics. Using the same test template gives the results presented in table 8 and the value *1.4* for step 1.

Table 8
Test results of 48 000 simulated test problems

| n | A | U1 | U2 | OPT |
|---|---|---|---|---|
| 5 | 5 | 991 | 955 | 1000 |
| | 10 | 997 | 981 | 1000 |
| | 15 | 1000 | 992 | 1000 |
| | 20 | 1000 | 989 | 1000 |
| 10 | 5 | 957 | 889 | 1000 |
| | 10 | 976 | 933 | 1000 |
| | 15 | 992 | 951 | 1000 |
| | 20 | 993 | 961 | 1000 |
| 15 | 5 | 921 | 821 | 1000 |
| | 10 | 949 | 902 | 1000 |
| | 15 | 981 | 922 | 1000 |
| | 20 | 986 | 940 | 1000 |
| 20 | 5 | 905 | 818 | 1000 |
| | 10 | 945 | 880 | 1000 |
| | 15 | 964 | 919 | 1000 |
| | 20 | 970 | 927 | 1000 |
| 25 | 5 | 889 | 800 | 1000 |
| | 10 | 940 | 863 | 1000 |
| | 15 | 945 | 904 | 1000 |
| | 20 | 960 | 892 | 1000 |
| 30 | 5 | 863 | 761 | 1000 |
| | 10 | 925 | 857 | 1000 |
| | 15 | 946 | 877 | 1000 |
| | 20 | 947 | 895 | 1000 |
| | Optimal solutions: | 44571(92.9%) | | |
| | Average error: | 0.008967 % | | |
| | Maximum error: | 1.49 % | | |

U1: D = U (100, 100 000), a = U (0.5, 5.0) and h = U (0.2, 3.0)
U2: D = U (100, 100 000), a = U (5.0, 17.5)) and h = U (0.2, 1.2)

In table 8, the U1 and U2 columns represent two different types of randomized values according to a uniform distribution. At both columns the numbers of optimal solutions are given that were reached with the heuristics.

Kaspi and Rosenblatt (1991) tested several heuristics. Our heuristic outperforms older heuristics such as Goyal (1973), Silver (1976), Goyal and Belton (1979), Goyal (1988b) and Kaspi and Rosenblatt (1983) in all three aspects (optimality, average and maximum error). In comparison with newer ones, Kaspi and Rosenblatt (1991) with the modification made by Goyal and Deshmukh (1993) our heuristic performs both better and worse. Comparing with Hariga (1994b) is not possible since he conducted his testing differently.

According to the test data made by Goyal and Deshmukh (1993), out of 48 000 test examples the modified RAND procedure produces 81.7 % optimal solutions, an average error of

0.00184 % and a maximum error of 0.381. As can be seen, our heuristic produces far more optimal solutions, but gives a slightly higher average and maximum error.

In general our method, as most methods works better when the major replenishment cost is relatively high. The most difficult problems to handle are when the major replenishment cost is lower than the minor replenishment costs.

As for the testing of which value of the quotients that gives the best performance, we have tested different values between one and two. If we look at formula (7) we can see that the largest possible decrease of a quotient, when the *m*-value is increased by one will be less than ¾ of the original value. This will happen when an *m*-value is increased from one to two. This means that if a quotient is two or higher an increase in the *m*-value will always give a lower total cost. Low values are not of interest since too many quotients will be put too low.

The results of all 48 000 simulated problems for different values is presented in table 9.

Table 9.
Test results of different values for step 1

| Value of quotients | Opt. solutions | Avr. error (%) | Max. error (%) |
|---|---|---|---|
| 1.0 | 43430 | 0.019207 | 2.46 |
| 1.1 | 43801 | 0.014879 | 2.46 |
| 1.2 | 44079 | 0.012286 | 1.60 |
| 1.3 | 44319 | 0.010455 | 1.72 |
| 1.4 | 44571 | 0.008967 | 1.49 |
| 1.5 | 44628 | 0.009864 | 2.01 |
| 1.6 | 44465 | 0.012653 | 3.23 |
| 1.7 | 44032 | 0.017468 | 5.24 |
| 1.8 | 43331 | 0.024049 | 5.24 |
| 1.9 | 42361 | 0.031075 | 5.24 |
| 2.0 | 42165 | 0.032089 | 5.24 |

When the major replenishment cost is low in comparison with the minor replenishment costs a lower value works better and vice versa. As seen from the results the value 1.5 gives most optimal solutions, but *1.4* gives a better average error and maximum error. Since the difference in optimal solutions between the two values is very small and the fact that the average- and maximum error are better we choose *1.4*. (Note, that using exactly $\sqrt{2}$ does not improve the results, only marginally worsen). It is clear that *1.4* it is not a critical value. Using close lying values would also work as well and only marginally change the results for the worse.

## 5. Conclusions

In this paper we present a new heuristic to solve the joint replenishment problem. It offers a different approach, a different way of thinking where we are not trying to determine the most favorable cycle times and order quantities instantly for all items in a recursion procedure as most earlier methods do.

Production planners, purchasers etc. in industry many times require access to Enterprise Resource Planning (ERP) software to deal with complex problems. As for the joint replenishment problem, the only "simple" method that has been presented is Silver (1976). This is also a famous method, which can be found in many books, possibly because of its simplicity. Unfortunately, the method performs in general worse than many newer methods, and could result in a high cost penalty (c.f. test data by Kaspi and Rosenblatt (1991)). The simplicity of different models, in terms of understanding and usage, is very difficult to measure. We still, however, argue that this new method is quite easy to understand and use. Since the model is based on a spreadsheet technique it is very suitable to use in spreadsheet programs on a routine basis. All this could be valuable for, probably primarily small- and medium size companies that don't have access to an ERP-software that can handle the JRP nor much knowledge of the problem. The usage in practical situations for a deterministic model is, however, limited since demand is stochastic in most situations.

From the extensive testing it is clear that our method performs well and outperforms older models. In comparison with the newer RAND procedure, our new heuristic produces a lot more optimal solutions, but gives a higher average- and maximum error. A small drawback is perhaps the maximum error, which ranged up to 1.49 %. One should, however, bear in mind that this was out of a very large test sample. The performance is still very good and it should hardly make a difference during practical use. Finally, the new heuristic solves joint replenishment problems in a novel way, and we believe that future modifications may increase the performance.

**Acknowledgement**

The authors are grateful for valuable comments and suggestions made by anonymous referees. This research has been supported by the supported by Jan Wallander and Tom Hedelius Research Foundations, project "J01-23".

**References**

Brown, R.G., 1967. Decision rules for inventory management, Holt, Reinhart and Winston, New York, 50-55.

Fung, RYK. and Ma, X., 2001. A new method for joint replenishment problems, Journal of the Operational Research Society 52, 358-362.

Goyal, S.K., 1973. Determination of economic packaging frequency of items jointly replenished, Management Science 20, 232-235.

Goyal, S.K., 1974. Determination of optimal packaging frequency of jointly replenished items. Management Science 21, 436-443.

Goyal, S.K. and Belton, A. S., 1979. On a simple method of determining order quantities in joint replenishments for deterministic demand. Management Science 25, 604.

Goyal, S.K., 1985. On improving Silver's algorithm for the joint replenishment problem. IIE Transactions 17, 99-101.

Goyal, S.K., 1988a. Determining the optimum production-packaging policy for jointly replenished items. Engineering Costs and Production Economics 15, 339-341.

Goyal, S.K., 1988b. Economic ordering policy for jointly replenished items. International Journal of Production Research 26, 1237-1240.

Goyal, S.K. and Deshmukh, S.G., 1993. A note on "the economic ordering quantity for jointly replenished items. International Journal of Production Research 31 (12), 2959-2961.

Hariga, M., 1994a. Two new heuristics procedures for the joint replenishment problem. Journal of the Operational Research Society 45(4), 463-71.

Hariga, M., 1994b. Comparative study of heuristics for the joint replenishment problem. Omega International Journal of Management Science 23(3), 341-344.

Kaspi, M. and Rosenblatt, M.J., 1983. An improvement of Silver's algorithm for the joint replenishment problem. IIE Transactions 15, 264-267.

Kaspi, M. and Rosenblatt, M.J., 1985. The effectiveness of heuristic algorithms for multi-item inventory systems with joint replenishment costs. International Journal of Production research 23, 109-116.

Kaspi, M. and Rosenblatt, M.J., 1991. On the economic ordering quantity for jointly replenished items. International Journal of Production Research, 29(1), 107-114.

Nocturne, D.J., 1973. Economic ordering frequency for several items jointly replenished. Management Science 19, 1093-1096.

Olsen, A.L., 2005. An evolutionary algorithm to solve the joint replenishment problem using direct grouping. Computers ad industrial Engineering 48, 223-235.

Silver, E.A., 1976. A simple method of determining order quantities in joint replenishments under deterministic demand. Management Science 22, 1351-1361.

Segerstedt, A., 1999. Lot sizes in a capacity constrained facility with available initial inventories. International Journal of Production Economics. 59, 469-475.

Van Eijs, M.J.G., 1993. A note of the joint replenishment problem under constant demand. Journal of the Operational Research Society 44(2), 185-91.

Viswanathan, S., 1996. A new optimal algorithm for the joint replenishment problem. Journal of the Operational Research Society 47, 936-44.

Viswanathan, S., 2002. On optimal algorithms for the joint replenishment problem. Journal of the Operational Research Society 53, 1286-1290.

Wildeman, R.E., Frenk, J.B.G. and Dekker, R., 1997. An efficient optimal solution method for the joint replenished problem. European Journal of Operational Research 99, 433-444.